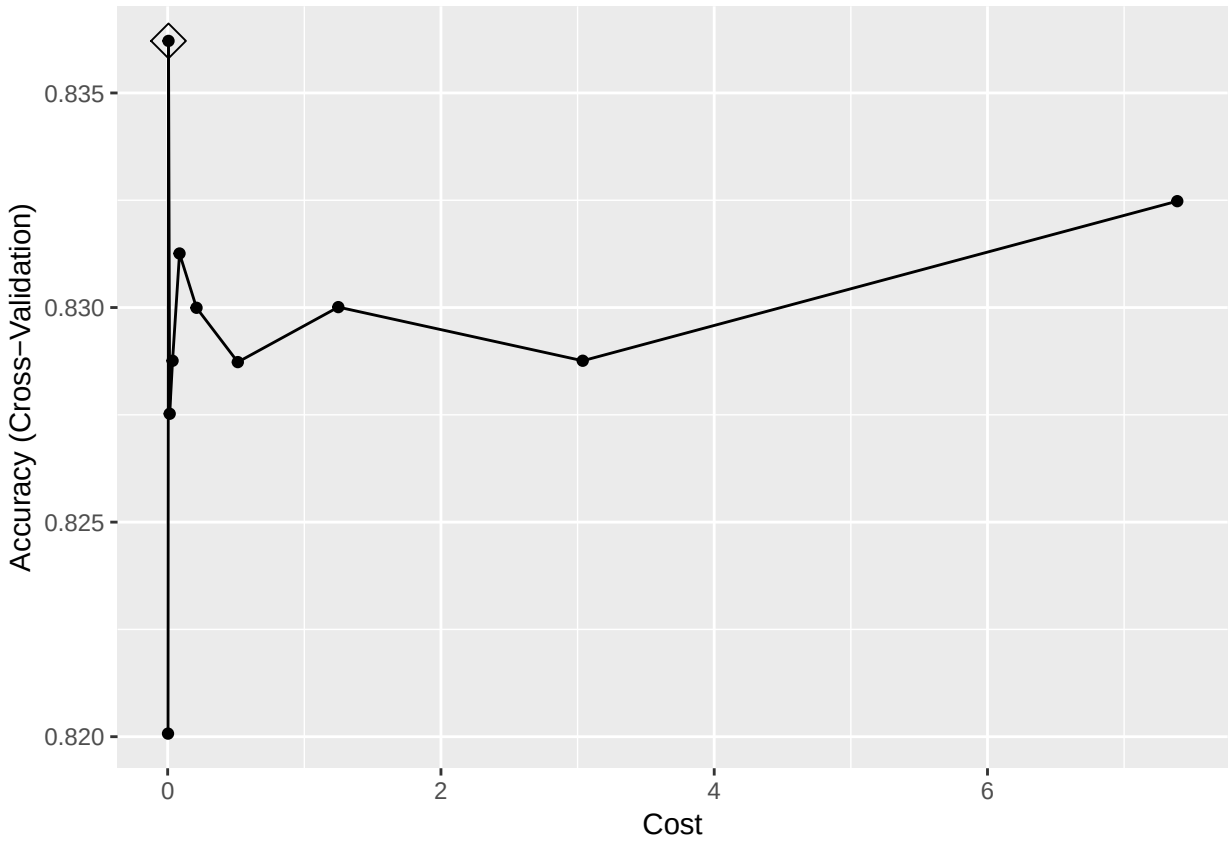
Homework 5

Deepika Dilip

Problem 1

```
data(OJ)
oj.data <- OJ

n = 799/1070
set.seed(1)
rowTrain <- createDataPartition(y = oj.data$Purchase, p=n, list = FALSE)

training <-oj.data[rowTrain,]
testing <-oj.data[-rowTrain,]

ctrl <- trainControl(method = "cv")

set.seed(1)
svml.fit <- train(Purchase~., data = training,
                  method = "svmLinear2",
                  preProcess = c("center", "scale"),
                  tuneGrid = data.frame(cost = exp(seq(-6,2,len=10))),
                  trControl = ctrl)

ggplot(svml.fit, highlight = TRUE)
```




```
#Training Error Rate
set.seed(1)
pred.svml.train <- predict(svml.fit, newdata = training)
cm.linear.train <-confusionMatrix(data = pred.svml.train, reference = training$Purchase)
train.error.linear = 1 - as.numeric(cm.linear.train$overall["Accuracy"])
print(train.error.linear)
```

```
## [1] 0.1675
```

```
#Test Error Rate
set.seed(1)
pred.svml.test <- predict(svml.fit, newdata = testing)
cm.linear.test <-confusionMatrix(data = pred.svml.test, reference = testing$Purchase)
test.error.linear = 1 - as.numeric(cm.linear.test$overall["Accuracy"])
print(test.error.linear )
```

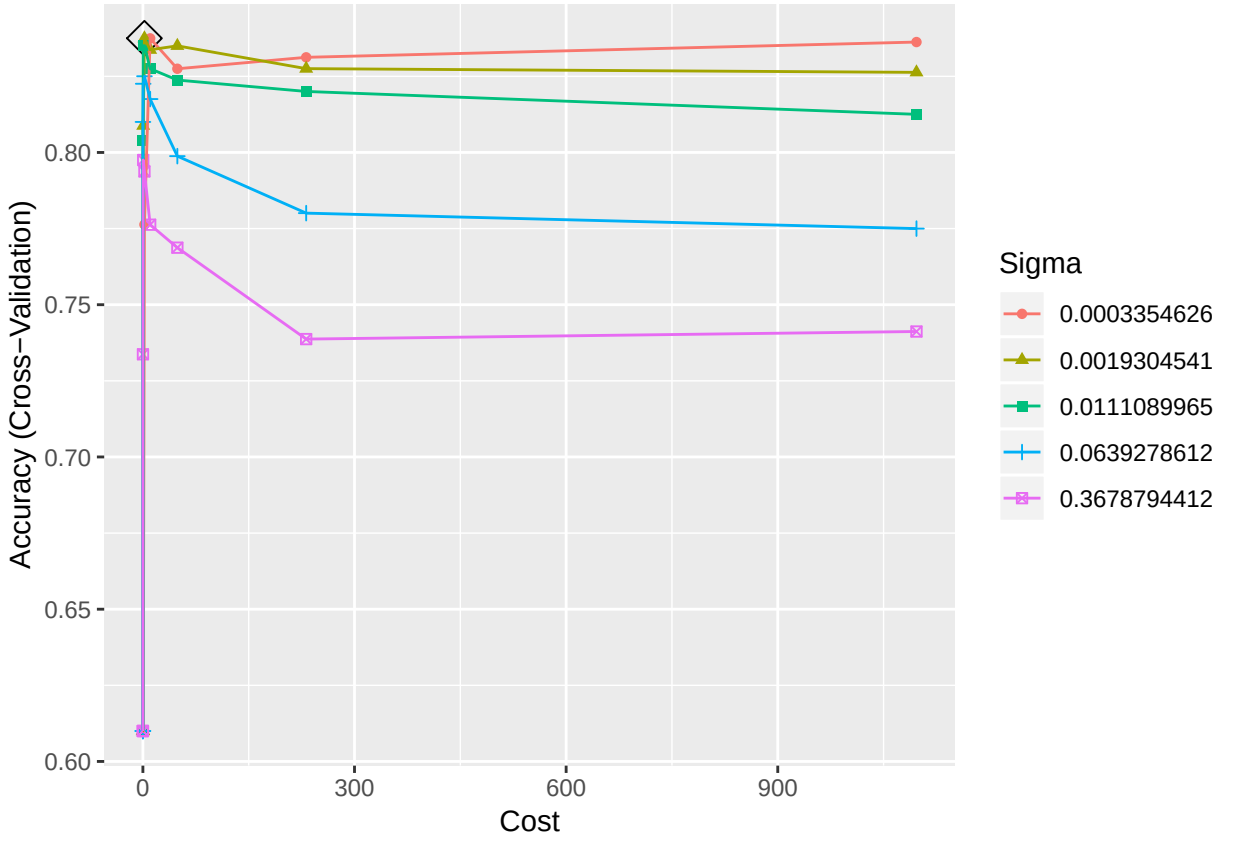
```
## [1] 0.1703704
```

The training error rate is 16.75% and the test error rate is 17.03%.

Problem 2

```
svmr.grid <- expand.grid(C = exp(seq(-7,7,len=10)),
                         sigma = exp(seq(-8,-1,len=5)))

set.seed(1)
svmr.fit <- train(Purchase~., data = training,
```

```
                    method = "svmRadial",
                    preProcess = c("center", "scale"),
                    tuneGrid = svmr.grid,
                    trControl = ctrl)

ggplot(svmr.fit, highlight = TRUE)
```




```
#Training Error Rate
set.seed(1)
pred.svmr.train <- predict(svmr.fit, newdata = training)
cm.radial.train <-confusionMatrix(data = pred.svmr.train, reference = training$Purchase)
train.error.radial = 1 - as.numeric(cm.radial.train$overall["Accuracy"])
print(train.error.radial)
```

```
## [1] 0.1625
```

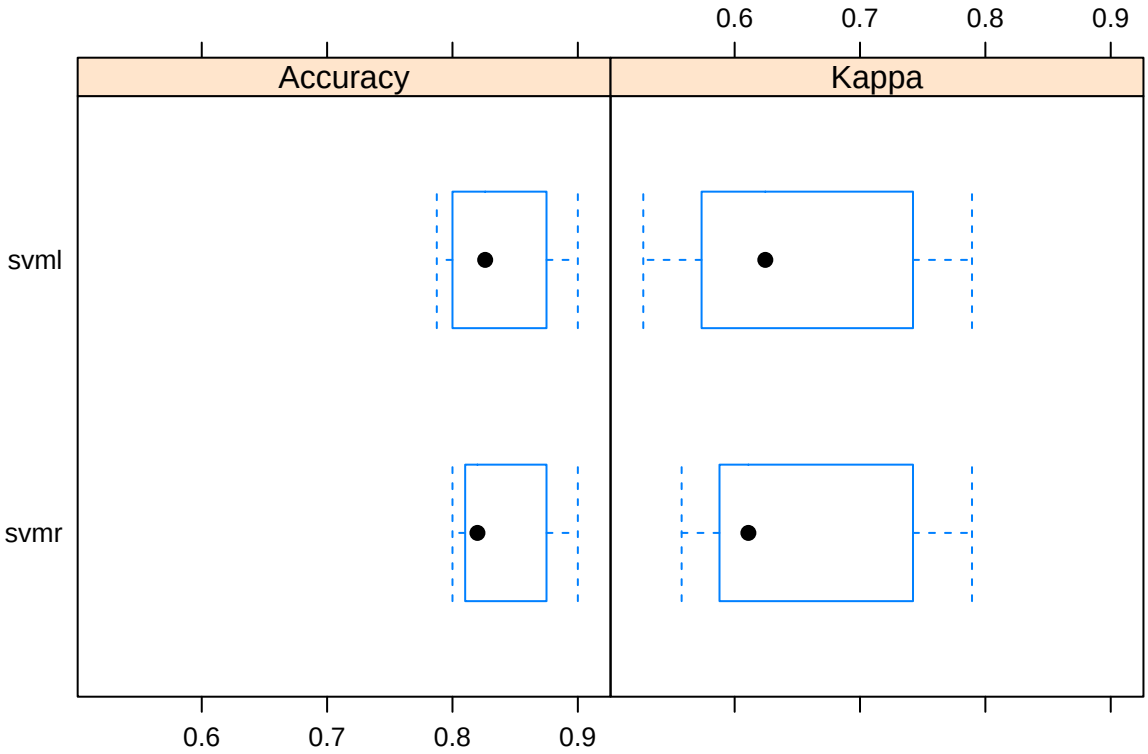
```
#Test Error Rate
set.seed(1)
pred.svmr <- predict(svmr.fit, newdata = testing)
cm.radial.test <-confusionMatrix(data = pred.svmr, reference = testing$Purchase)
test.error.radial = 1 - as.numeric(cm.radial.test$overall["Accuracy"])
print(test.error.radial)
```

```
## [1] 0.1666667
```

The training error rate is 16.25% and the test error rate is 16.67%.

Problem 3

```
resamp <- resamples(list(svmr = svmr.fit, svml = svml.fit))
bwplot(resamp)
```




```
summary(resamp)
```

```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: svmr, svml
## Number of resamples: 10
##
## Accuracy
##        Min.   1st Qu.    Median      Mean  3rd Qu. Max. NA's
## svmr 0.8000 0.8107199 0.8199074 0.8374941 0.865625  0.9    0
## svml 0.7875 0.8031250 0.8260802 0.8362129 0.865625  0.9    0
##
## Kappa
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## svmr 0.5577056 0.5891070 0.6109570 0.6507882 0.7206083 0.7893351    0
## svml 0.5271210 0.5771912 0.6244577 0.6481967 0.7206083 0.7893351    0
```

From the box-plots and the resampling, we can conclude that the linear kernel has a slightly higher accuracy. Therefore, we should use the linear kernel approach.